Tagging Stack Overflow Questions

A multi-class problem

The Problem

17m+

Questions on Stack Overflow

54k+

Tags on Stack Overflow

Challenges Deep-Dive

Number of tags

Over 54k tags

When the number of tags is increased, the odds of a proper selection are decreased.

Types of tags

Similar tag vocabulary

Many programming languages contain similar words or phrases

Filtering Text

Data in html format

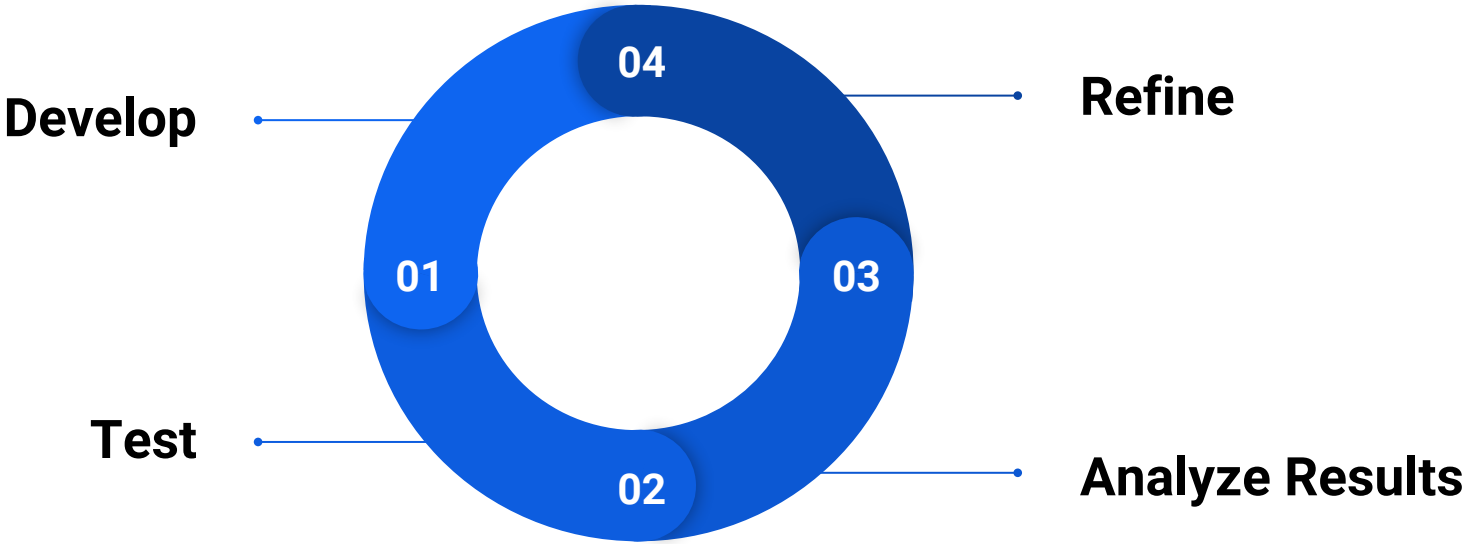
The data was in an html format and included non-alphanumeric characters

In addition, challenge throughout to find words most related to tag

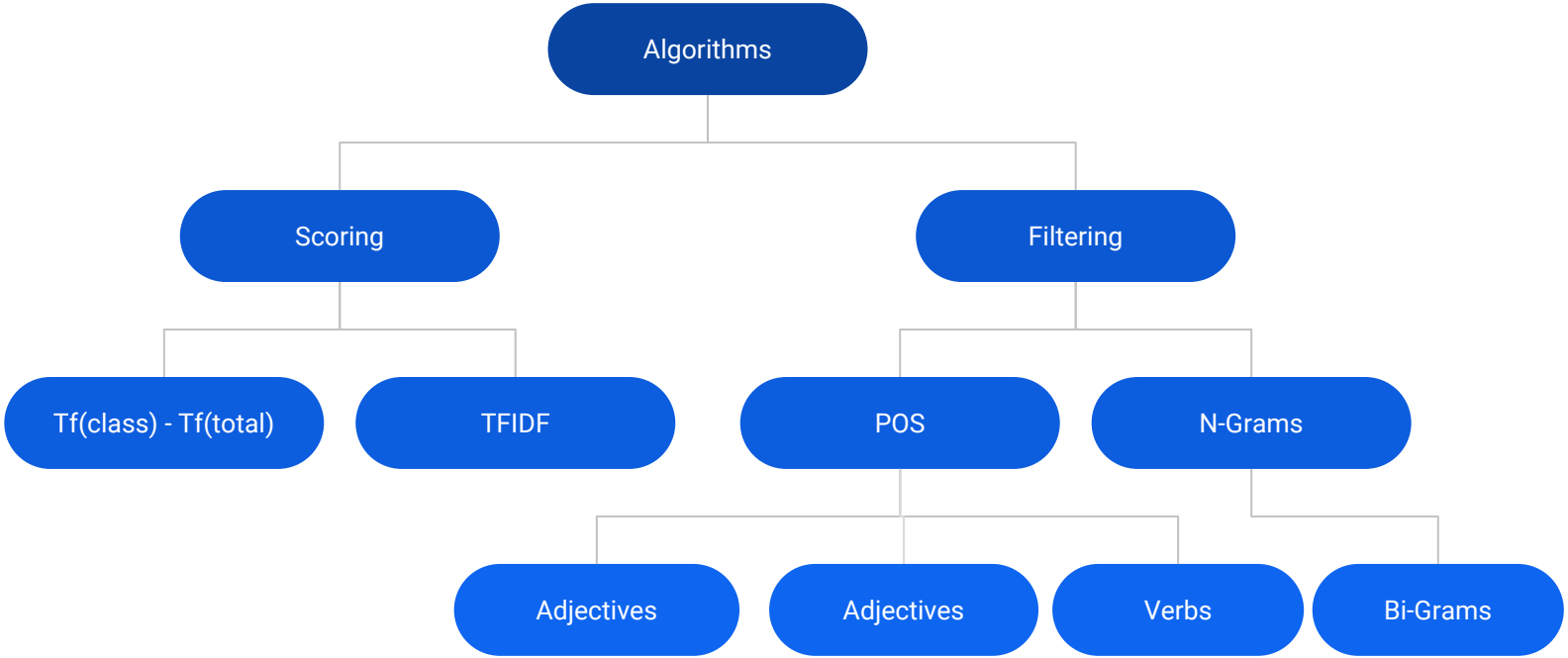
Solutions

Development Cycle

How to walk through solutions



The Algorithms



Naive Bias

Baseline for the rest of testing

21%

Naive Bias

Simple Naive Bias to get a baseline for future test

- Filler words often determine guessed tag

54%

Naive Bias with stop words filtered

Filtering stop words flushing out important words

- Increased accuracy
- Higher look at important words in tagging

POS Tagging

Filtering Method

74%

Nouns

Filtering out text so that only nouns are analyzed

- Nouns are good indication of overall subject

47%

Adjectives

Filtering out text so that only adjectives are analyzed

- Adjectives across different programming languages can be very similar

55%

Verbs

Filtering out text so that only verbs are analyzed

- Verbs across different possible tags are not very distinguished

Bi-Grams

Filtering Method

62%

Bi-Grams

Use Naive Bias scoring on Bi-grams in questions

- Bi-grams produce more unique words
- Bi-grams are more informative on the type of tag overall

Scoring Algorithms

2%

Tf(class) - Tf(total)

Take the term frequency per class and subtract by term frequency in total test docs

- Terms are not as unique as expected across tags
- Filler words still determining tag

71%

Tfidf

Term frequency times inverse document frequency

- Heavily scored infrequent terms

Scoring and filtering (POS + tfidf)

70%

POS + tfidf

Applying tfidf to only the nouns in a question

- No overall improvement
- Increase training size
- Tags very similar

Filtering

Filler words can completely throw off an NLP algorithm, while proper filtering can give surprising improvements

Weighing

Scoring words by their uniqueness to the tag can help improve tagging accuracy but comes with challenges

Improve

Filter out by tri-grams
Bi-gram + Tfidf
POS with noun phrases

Questions?

Assessing Toxicity in Wikipedia Comments

Jonathan Innis & Gabriel Britain

Disclaimer: Some comments in this presentation may be offensive to certain viewers. The comments in this presentation do not reflect the opinions of the creators/presenters and are used purely for academic purposes.

Purpose

- Identifying toxicity can prevent users from abusing communication platforms
- Much more efficient than review by human moderators
- Most comments are posted at early hours of the morning (3am) and will be uncaught by human moderators for hours

How to Deal With a $759 Million Lottery Jackpot

Toxic

Carl Hollis · 2 hours ago

two ignorant idiots you two are.

↑0 ↓0 Edit Reply

Approve Spam Delete

Arpaio Pardon Would Show Contempt for Constitution

OnKilter · 5 hours ago

STFU racist pig.
Just STFU.

↑0 ↓0 Edit Reply

Approve Spam Delete

92% similar to comments people said were "toxic" SEEM WRONG?

Nieman Lab is a great website — only an idiot like you would think some other website could possibly be better. You dumb jerk.

[redacted] Nov 30, 2014 +2

First of all, A for effort! But I wasn't a racist [redacted] like you were, so my grammar is irrelevant (so I'm not a hypocrite, although that's a big word, you should be proud). Also, I should point out that yours didn't improve, so we got nowhere with you. Your spelling makes me inclined to think you're a 'dirty [redacted] yourself! And I hope at the end that you weren't threatening to kill me. I'll forgive you because you seem cranky, so I'd suggest a nap, you mouth-breathing, stagnant cesspool of human trash.

Show less

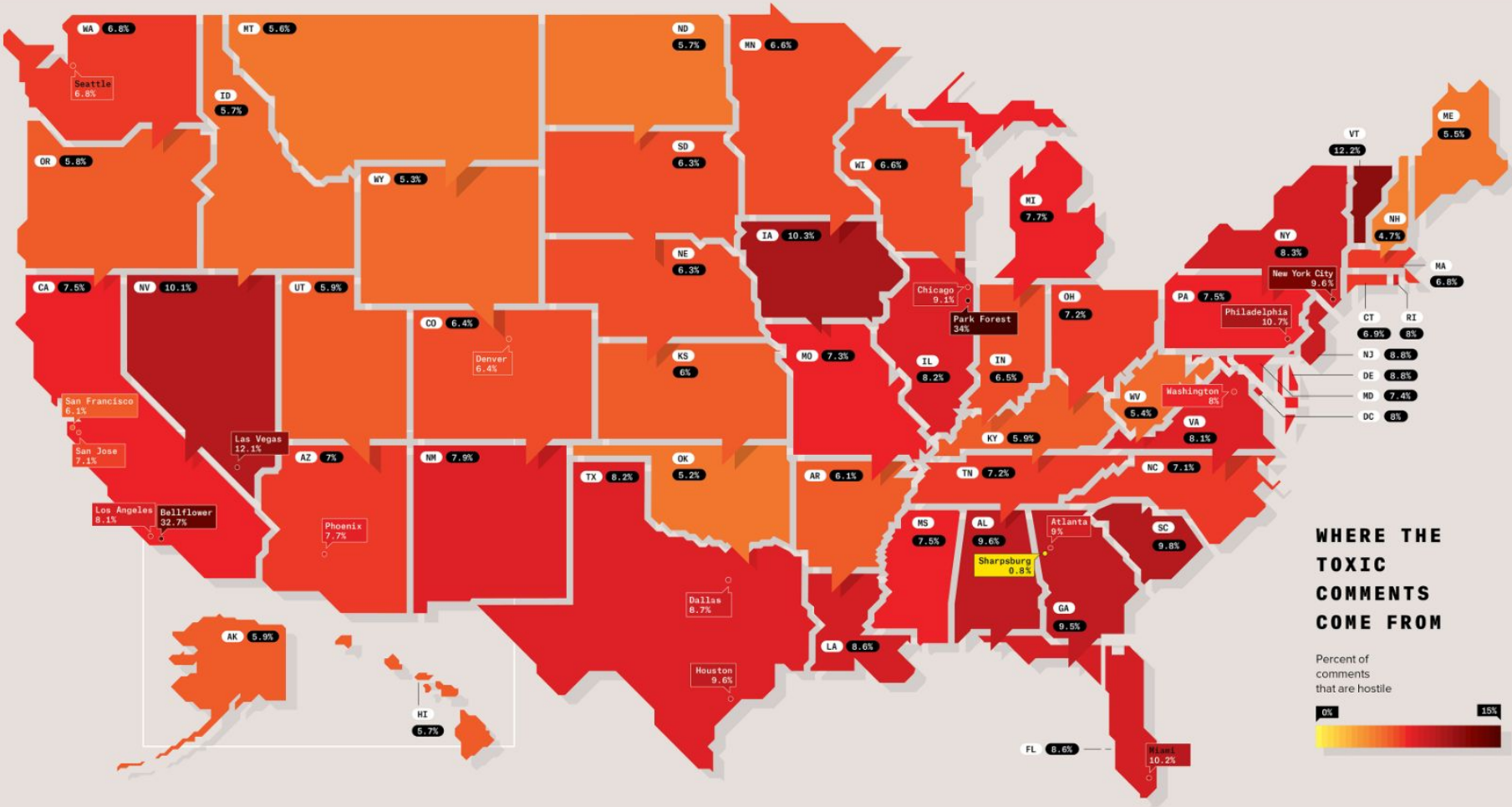
98% similar to comments people said were "toxic" SEEM WI

You're a stupid idiot!

UnknownArchive 1 week ago

#1 [redacted] this feminist [redacted] and all the damage she did to thatguywiththeglasses.
#2 see #1

Reply · 15



https://www.wired.com/2017/08/internet-trol
/

Dataset Source

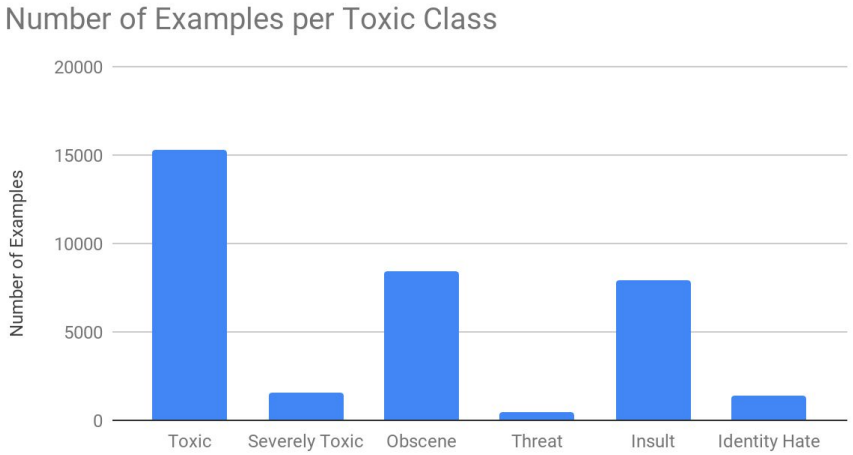
id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0000997932d777bf	Explanation Why the edits made under my username Hardcore Metallica Fan were	0	0	0	0	0	0
000103f0d9cfb60f	D'aww! He matches this background colour I'm seemingly stuck with. T	0	0	0	0	0	0
000113f07ec002fd	Hey man, I'm really not trying to edit war. It's just that this guy is con	0	0	0	0	0	0
0001b41b1c6bb37e	" More I can't make any real suggestions on improvement - I wondered if the s There appears to be a backlog on articles for review so I guess there n	0	0	0	0	0	0
0001d958c54c6e35	You, sir, are my hero. Any chance you remember what page that's on?	0	0	0	0	0	0
00025465d4725e87	" Congratulations from me as well, use the tools well. · talk "	0	0	0	0	0	0
0002bcb3da6cb337	[redacted] BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
00031b1e95af7921	Your vandalism to the Matt Shirvington article has been reverted. Plea	0	0	0	0	0	0
00037261f536c51d	Sorry if the word 'nonsense' was offensive to you. Anyway, I'm not inter	0	0	0	0	0	0
00040093b2687caa	alignment on this subject and which are contrary to those of DuLithgow	0	0	0	0	0	0
0005300084f90edc	" Fair use rationale for Image:Wonju.jpg Thanks for uploading Image:Wonju.jpg. I notice the image page specifi Please go to the image description page and edit it to include a fair use If you have uploaded other fair use media, consider checking that you h Unspecified source for Image:Wonju.jpg Thanks for uploading Image:Wonju.jpg. I noticed that the file's descripti As well as adding the source, please add a proper copyright licensing t If you have uploaded other files, consider checking that you have speci	0	0	0	0	0	0
00054a5e18b50dd4	bbq be a man and lets discuss it-maybe over the phone?	0	0	0	0	0	0

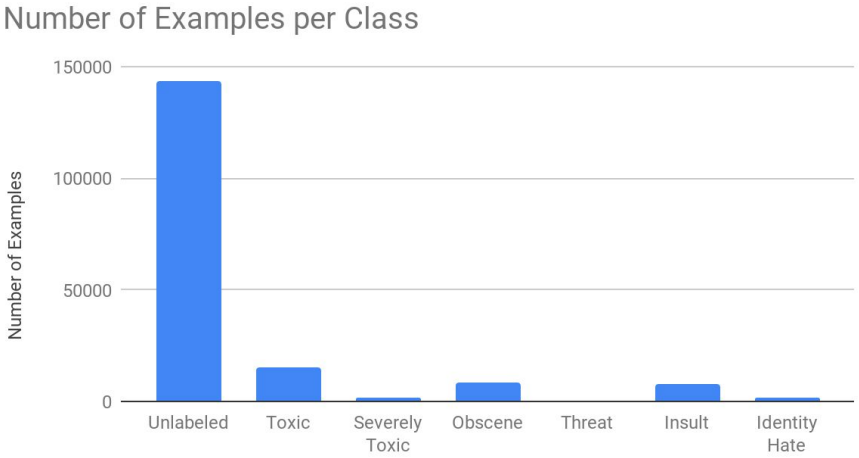
Frameworks

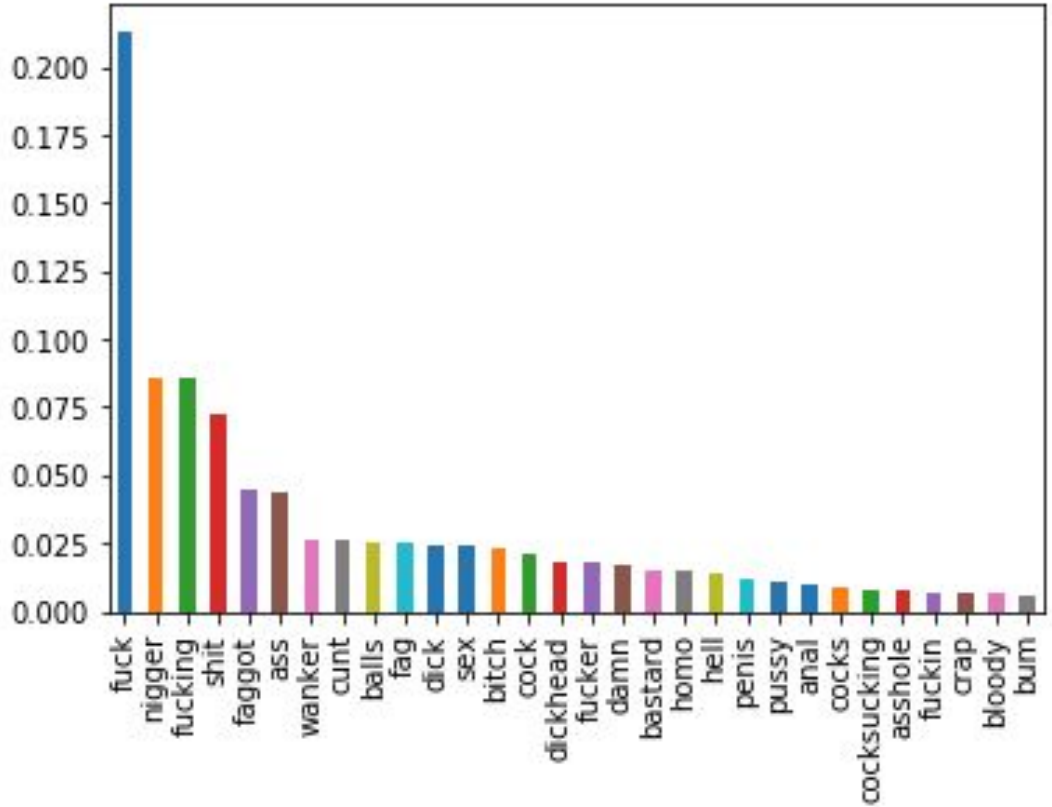
Data Inspection

1. Class Distribution
2. Common Toxic Word Inspection
3. Comment Length Inspection

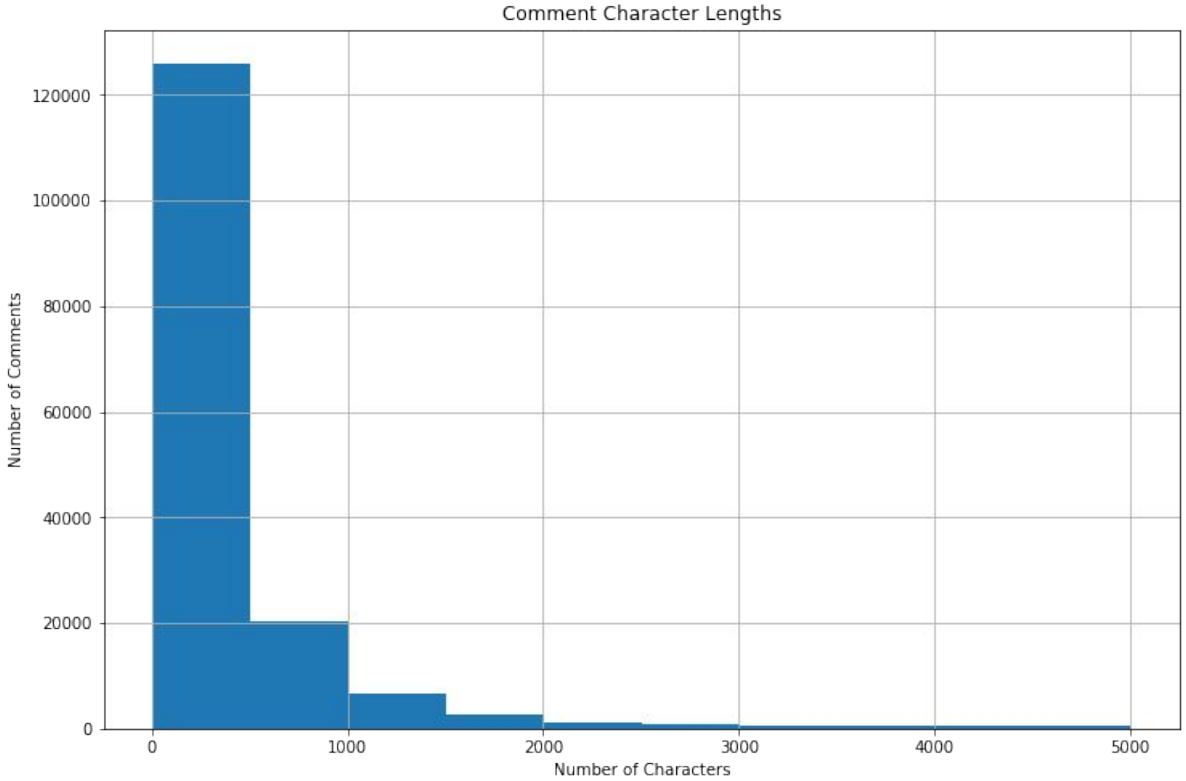
Class Distribution







Common Toxic Words



Comment Character Lengths

Baseline

Random Assignment (based on class frequencies)

	Precision	**Recall**	**F1-Score**	**Support**
Toxic	0.10	0.43	0.16	5038
Severely Toxic	0.01	0.06	0.02	500
Obscene	0.05	0.23	0.08	2810
Threat	0.00	0.02	0.01	152
Insult	0.05	0.23	0.08	2591
Identity Hate	0.01	0.04	0.01	449
Micro Avg	0.07	0.30	0.11	11540
Macro Avg	0.04	0.17	0.06	11540
Weighted Avg	***0.07***	***0.30***	***0.11***	***11540***

Models

1. Naive Bayes Classifier
2. Support Vector Machines
3. Random Forest Classifier
4. Recurrent Neural Network

Naive Bayes Classifier

- “Bag of Words” model makes sense for toxic comment classification
- Precision, Recall, & F1 strong improvements over baseline

	Precision	**Recall**	**F1-Score**	**Support**
Toxic	0.83	0.59	0.69	5042
Severely Toxic	0.31	0.79	0.44	557
Obscene	0.78	0.79	0.79	2761
Threat	0.05	0.78	0.09	163
Insult	0.65	0.68	0.66	2623
Identity Hate	0.19	0.58	0.29	481
Micro Avg	0.53	0.67	0.59	11627
Macro Avg	0.47	0.70	0.49	11627
Weighted Avg	***0.71***	***0.67***	***0.67***	***11627***

Feature Analysis

- Naive Bayes found certain features (unigrams, bigrams, and trigrams) that are most useful to the model

toxic:
- 2123145146
- kundad
- kunstruktive
- kunt
- kupla
- kurang
- yammer
- follarte
- fuckyourself
- crackhead

severe_toxic:
- stomes
- stikin
- caspa
- anastal1111you
- ancest
- ancestryearly
- ancestryerigate
- ada_at
- cartuchos
- homelan

obscene:
- achivements
- achmed
- achsehole
- kcik
- sexmist
- britch
- britbarb
- katzrin
- zigabo
- follarte

threat:
- m45terbate
- ma5terb8
- ma5terbate
- master-bate
- masterb8
- masterbat*
- masterbat3
- teeeccccctoooooniiiiiiccccccc
- hawkinghttp
- zigabo

insult:
- faggots129
- islantic
- snigbrook
- furfag
- fortuijn
- 66185192207
- libtard
- onanizing
- crackhead
- suberbia

identity_hate:
- gomnna
- closerlookonsyria
- nawmean
- goddammed
- clubz
- goains
- nebracka
- negrate
- uos
- zigabo

Support Vector Machines

- Word embeddings to produce embeddings for each sentence
- Leveraged GloVe embeddings
- Leveraging custom embeddings could produce better results with greater resources and greater time

	Precision	Recall	F1-Score	Support
Toxic	0.96	0.06	0.12	6090
Severely Toxic	0.00	0.00	0.00	367
Obscene	0.95	0.09	0.16	3691
Threat	0.00	0.00	0.00	211
Insult	0.67	0.01	0.03	3427
Identity Hate	0.00	0.00	0.00	712
Micro Avg	0.93	0.05	0.10	14498
Macro Avg	0.43	0.03	0.05	14498
Weighted Avg	***0.80***	***0.05***	***0.10***	***14498***

Random Forest Classifier

- Resistant to class imbalance
- Decent results that suffered in the macr average performing poorly in the smaller classes

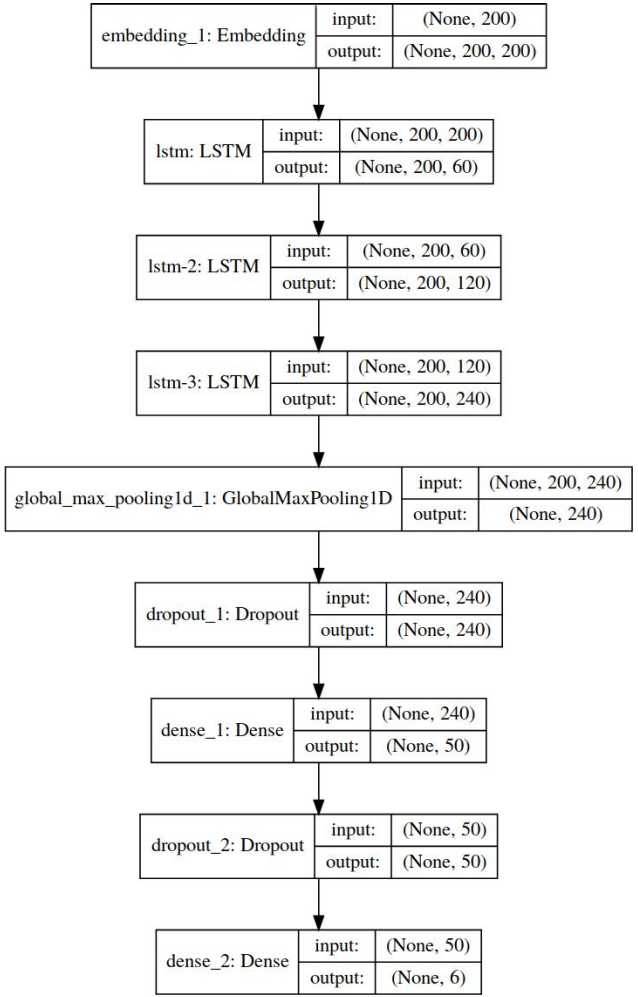
	Precision	Recall	F1-Score	Support
Toxic	0.57	0.76	0.65	6090
Severely Toxic	0.23	0.08	0.12	367
Obscene	0.58	0.68	0.63	3691
Threat	0.33	0.05	0.09	211
Insult	0.56	0.52	0.54	3427
Identity Hate	0.57	0.12	0.20	712
Micro Avg	0.57	0.62	0.59	14498
Macro Avg	0.47	0.37	0.37	14498
Weighted Avg	***0.56***	***0.62***	***0.57***	***14498***

Recurrent Neural Network (RNN)

- LSTMs shown to effectively handle long sequence
- Captures sentence structure

	Precision	Recall	F1-Score	Support
Toxic	0.57	0.85	0.68	6090
Severely Toxic	0.34	0.48	0.40	367
Obscene	0.60	0.80	0.68	3691
Threat	0.00	0.00	0.00	211
Insult	0.52	0.72	0.61	3427
Identity Hate	0.67	0.22	0.34	712
Micro Avg	0.56	0.75	0.64	14498
Macro Avg	0.45	0.51	0.45	14498
Weighted Avg	***0.56***	***0.75***	***0.63***	***14498***

RNN Architecture



Attributions

[1] Mart´ın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Man´e, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vi´egas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Ga¨el Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pages 108–122, 2013.

[3] Fran¸cois Chollet et al. Keras. https://keras.io, 2015.

[4] J. D. Hunter. Matplotlib: A 2d graphics environment. Computing In Science & Engineering, 9(3):90–95, 2007.

[5] Wes McKinney. Data structures for statistical computing in python. In St´efan van der Walt and Jarrod Millman, editors, Proceedings of the 9th Python in Science Conference, pages 51 – 56, 2010.

[6] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, 2014.

[7] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In Proceedings of the 26th International Conference on World Wide Web, WWW '17, pages 1391–1399, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

AGGRO

Declarative Programming in Natural Language

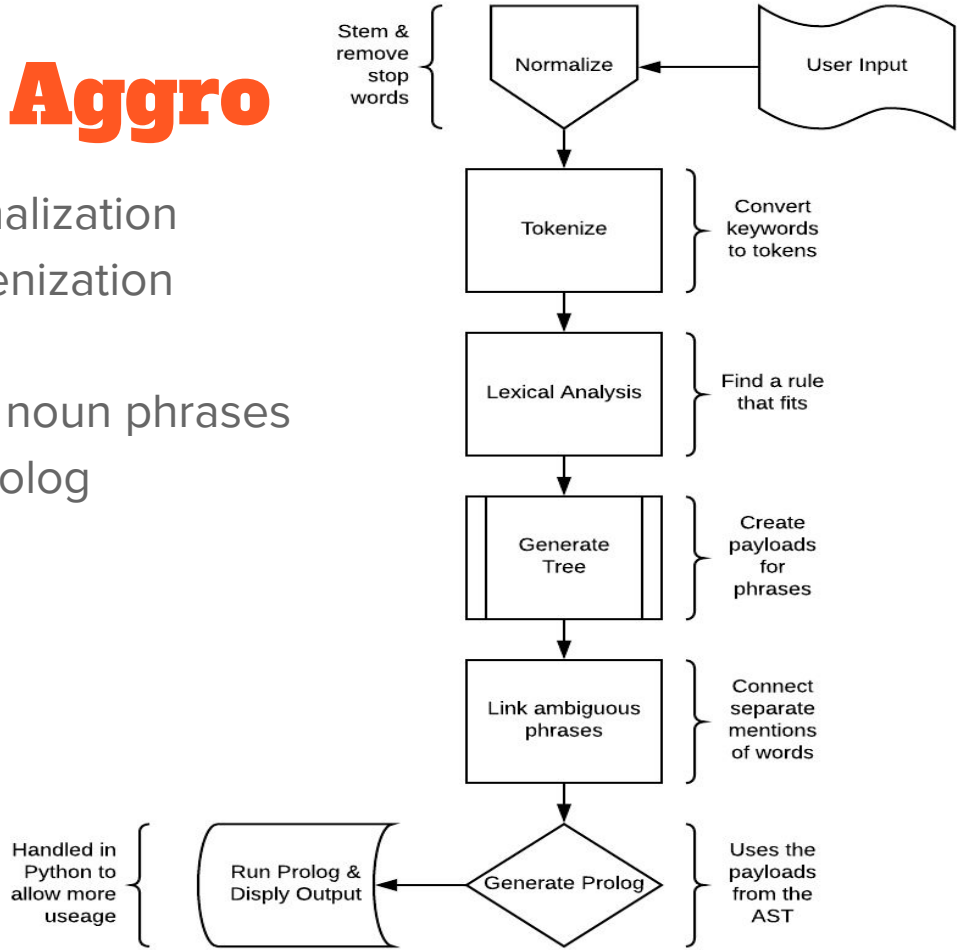
Ryan Beltran
Joseph Gerules

Description & Examples

- Aggro answers questions that are phrased in standard english.
- Examples:
  - A year is wild if and only if 2 divides the year evenly. The year is 2018. Is the year a wild one? -**TRUE**
  - A number n is prime if there exists no number m in the range of 1 to n such that m divides n evenly. Is 73 prime? - **TRUE**
  - "If and only if there is rain then there is water. There is not rain. Is there water?" - **FALSE**

The Five Phases of Aggro

1. Perform Stemming and Text Normalization
2. Perform Lexical Analysis and Tokenization
3. Generate Abstract Syntax Tree
4. Analyze and correlate ambiguous noun phrases
5. Generate, execute, and display Prolog



Stage 1 - Preprocessing

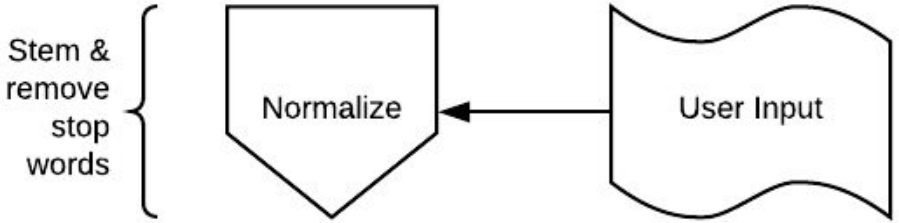
- Utilize Python's natural language toolkit, NLTK, library for:
  - POS tagging
  - Stemming
  - Remove stop words
  - Lowercasing
  - Tokenize based on POS tagging

Input:

A year is wild if and only if 2 divides the year evenly. The year is 2018. Is the year a wild one?

Output:

a year is wild if and onli if 2 divides the year evenli . the year is 2018. is the year a wild one ?

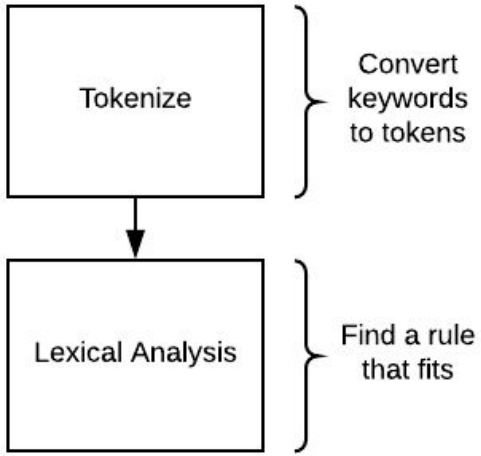


Stage 2 - Dynamic Tokenization

- Use Lex to parse the now preprocessed input
- Categorize & catch words to assign labels to them
  - Reserved words like “is” or “equals” get tagged as ‘EQUALS’

Input:
a year is wild
Output:

```
LexToken(A,'a',1,0)
LexToken(UNWORD,'year',1,2)
LexToken(EQUALS,'is',1,7)
LexToken(UNWORD,'wild',1,10)      #UNWORD is short for uniqueword
```



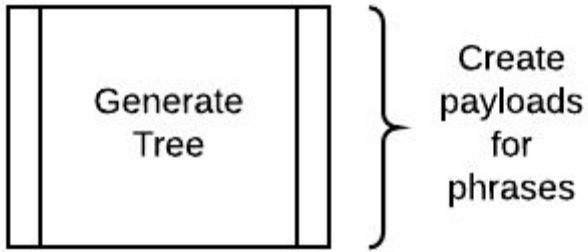
Stage 3 - Abstract Syntax Tree Generation

- Use yacc to parse the lexemes
  - Words or phrases turn to labeled nodes
  - Rules decide labels and connection order

- Considered Approaches:
  - GREMLIN: Levenshtein Optimal Fuzzy Grammars
  - Maximum Entropy Classification: Trained rule based classifier

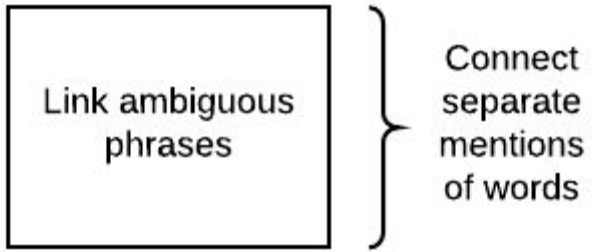
a year is wild if and onli if 2 divides the year evenli .

```
Node: __program__ [17]
| Node: __rule__ [16]
| | Node: __iff then__ [15]
| | | Node: __if__ [13]
| | | | Node: __is__ [12]
| | | | | Node: __modulo__ [9]
| | | | | | Phrase: { alias:, bound:False } [8]
| | | | | | | Leaf: year [7]
| | | | | | Node: __numeric const__ [6]
| | | | | | | Leaf: 2 [5]
| | | | | Node: __numeric const__ [11]
| | | | | | Leaf: 0 [10]
| | | Node: __then__ [14]
| | | | Node: __is__ [4]
| | | | | Phrase: { alias:, bound:False } [1]
| | | | | | Leaf: year [0]
| | | | | Phrase: { alias:, bound:False } [3]
| | | | | | Leaf: wild [2]
```



Stage 4 - Phrase Analysis

- Seeks to connect correlated phrases
- Tasks:
  - Correlate related phrases in an alias table
  - Properly split adjacent noun phrases
  - Label phrases as free or bound
- Phrase correlation based on two part metric:
  - Levenshtein similarity metric
  - Bayesian probability metric
- Split phrases to maximize total similarity
- How do we handle the word “it”?



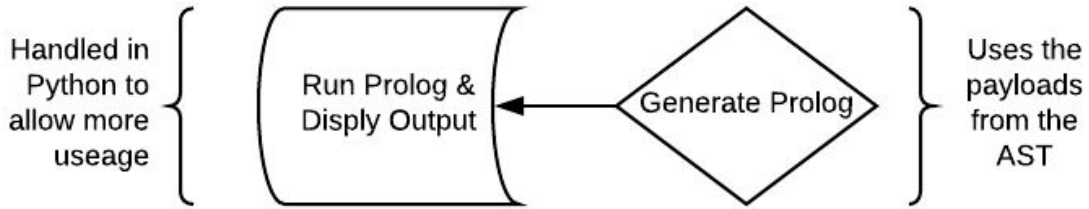
Stage 5 - Code Generation & Execution

- Use the AST's labels & node structure to write generic Prolog functions.
  - Add id's to each label to ensure uniqueness of generically named functions
  - Push all generated rules into the query statement to create a scope
- Use SWIPL to call SWI Prolog from Python.

```
water is wet . is water wet ?
is_10( A, A ).        // water is wet - Question form
is_4( A, A ).         // water is wet - Rule Form

query_11(  ) :- is_4( Phrase_0, Phrase_1 ), is_10( Phrase_0, Phrase_1 ). /*11*/ // Adding both rules creates a scope

query_11(  )  is:
true
```



Future Development

- Produce more grammar rules for a more robust system.
  - This would allow for more edge cases to be handled
  - Different styles of questions could be added
- Integrate pronoun and ambiguous word binding more thoroughly.
  - Look for nearby nouns.
- Make the outputted code more readable
  - Formatting
  - Implement using attributed objects instead of rules
- Fuzzy grammars to handle oddly worded input
- Improved phrase splitting
  - Is the grey cat very large?
    - Is the grey | cat very large or Is the grey cat | very large or Is the grey cat very | large
- Improved handling of free variables
  - The year y is a leap year if it is divisible by 4.
    - Y isn't a specific year. It is an unbounded free variable.

Math Question Answering (SemEval Task 10)

Kevin Sittser

The Problem

- **Closed-vocabulary algebra**, e.g. "Suppose 3x + y = 15, where x is a positive integer. What is the difference between the largest possible value of y and the smallest possible value of x, assuming that y is also a positive integer?"
- **Open-vocabulary algebra**, e.g. "At a basketball tournament involving 8 teams, each team played 4 games with each of the other teams. How many games were played at this tournament?"
- **Geometry**, e.g. "The lengths of two sides of a triangle are (x-2) and (x+2), where x > 2. Which of the following ranges includes all and only the possible values of the third side y?"

- Success metric: Percentage of problems solved

My Approach

- Ignore diagrams
- Equation solver!
- ~~Dimensional analysis??~~
- ~~Write a parser??~~
- Ignore wordy problems
- SymPy
- Algorithms??

Program Structure

- Question parser: Look for equations
- Equation parser: Convert to SymPy-readable (if possible)
- Solver: Equations, expressions → Result!
- Find closest valid answer
- (If can’t solve problem, output “C” (or “5”))

Complications

- Training?
- SymPy solve() output
- SymPy can’t handle everything
- So many equation formats

Results

- 20.82% correct answers!
- But a random guesser solves 19.38%
- Not very good
- Only target problems: 22.70% (vs. 19.03%) → a little better

Potential Improvements

- Training program?
- Write my own labels?
- Make sure all equations are LaTeX
- Accept more operation types (SymPy research)
- Syntactical analysis??
- Long, long process

Questions?